MATH 545 - Assignment 4

Emir Sevinc 260682995

November 29, 2018

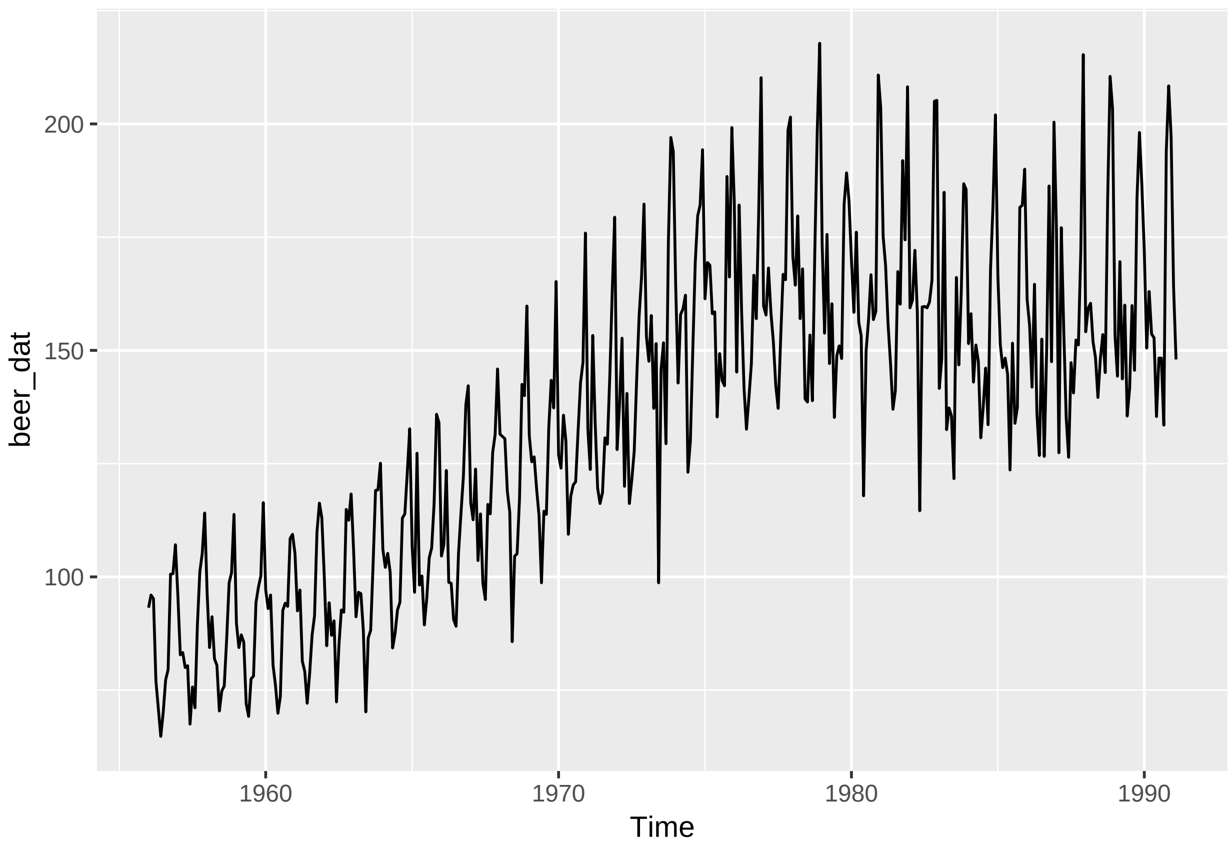
```r
library(tidyverse)
library(itsmr)
library(forecast)
library(tibbletime)
library(tsbox)
library(gridExtra)
library(TTR)
library(tidyquant)
library(here)
library(fpp2)
library(tseries)
```

6.9)

a)

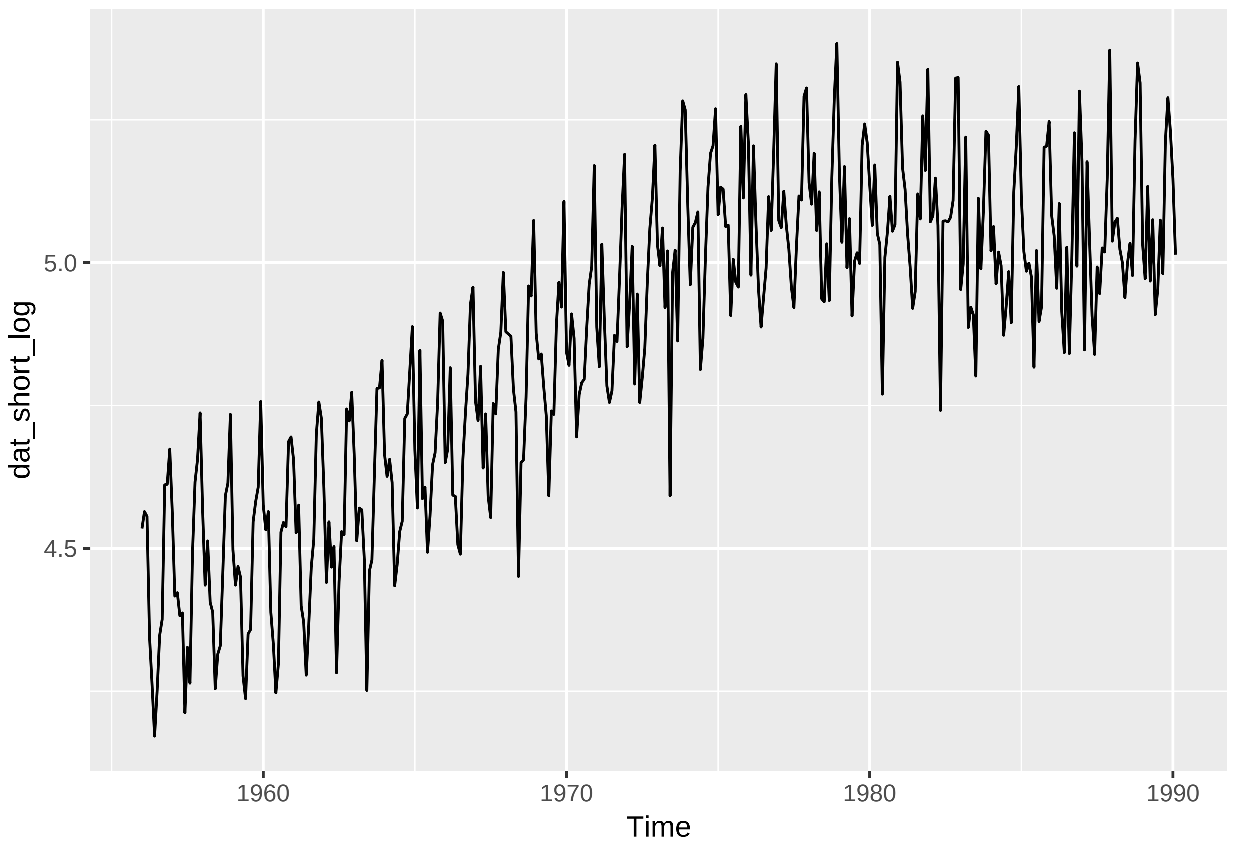
First we take a look at our data:

```r
beer_dat = dget("beer.Rput")
autoplot(beer_dat)
```



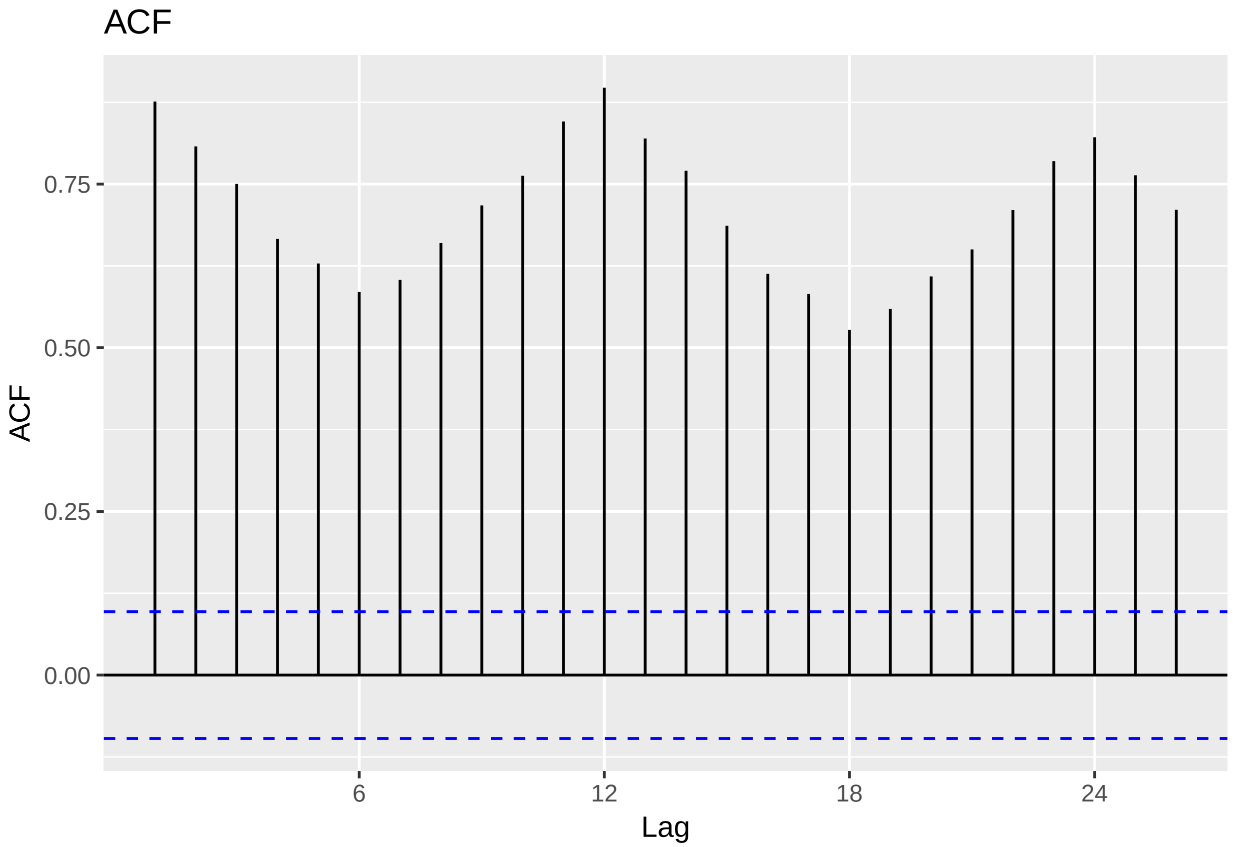
And below is the shortened data with the last 12 entries removed, and logged as instructed:

```
dat_short = dget("BSHORT.Rput")
dat_short_log <- log(dat_short)
autoplot(dat_short_log)
```

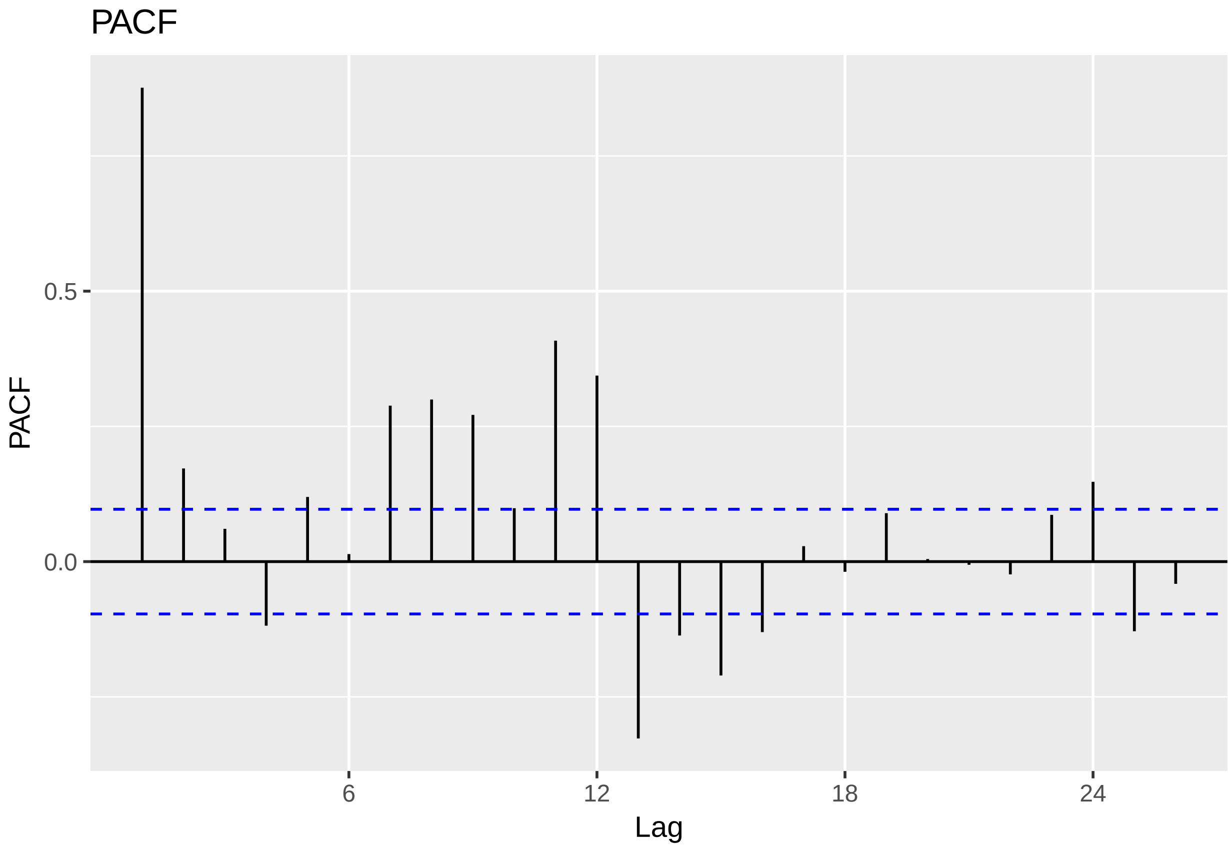


We now take a look at the ACF and PACF

```
ggAcf(dat_short_log) + ggtitle("ACF")
```



```
ggPacf(dat_short_log) + ggtitle("PACF")
```



The spikes at lags 12 and 24 suggest the usage of differencing, so we will set seasonal=TRUE.

Below we select the most appropriate model:

```
auto_select_AIC = auto.arima(dat_short_log,stepwise=FALSE,seasonal=TRUE,ic="aic",trace=TRUE,approximati
```

```
##
##  ARIMA(0,1,0)(0,1,0)[12]                    : -489.8496
##  ARIMA(0,1,0)(0,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,0)[12]                    : -537.5637
##  ARIMA(0,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,0)[12]                    : -553.6624
##  ARIMA(0,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,2)[12]                    : -700.0184
##  ARIMA(0,1,1)(0,1,0)[12]                    : -808.1
##  ARIMA(0,1,1)(0,1,1)[12]                    : -950.9193
##  ARIMA(0,1,1)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,0)[12]                    : -853.2381
##  ARIMA(0,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(2,1,0)[12]                    : -874.7052
##  ARIMA(0,1,1)(2,1,1)[12]                    : -960.0448
##  ARIMA(0,1,1)(2,1,2)[12]                    : -966.6671
##  ARIMA(0,1,2)(0,1,0)[12]                    : -820.2853
##  ARIMA(0,1,2)(0,1,1)[12]                    : -964.2873
##  ARIMA(0,1,2)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,2)(1,1,0)[12]                    : -868.821
##  ARIMA(0,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,2)(1,1,2)[12]                    : -968.0854
##  ARIMA(0,1,2)(2,1,0)[12]                    : -897.9965
##  ARIMA(0,1,2)(2,1,1)[12]                    : -971.4001
##  ARIMA(0,1,3)(0,1,0)[12]                    : -823.7839
##  ARIMA(0,1,3)(0,1,1)[12]                    : -971.5622
##  ARIMA(0,1,3)(0,1,2)[12]                    : -976.6105
##  ARIMA(0,1,3)(1,1,0)[12]                    : -874.4946
##  ARIMA(0,1,3)(1,1,1)[12]                    : -975.4362
##  ARIMA(0,1,3)(2,1,0)[12]                    : -902.538
##  ARIMA(0,1,4)(0,1,0)[12]                    : -826.4736
##  ARIMA(0,1,4)(0,1,1)[12]                    : -971.0385
##  ARIMA(0,1,4)(1,1,0)[12]                    : -878.4321
##  ARIMA(0,1,5)(0,1,0)[12]                    : -825.5407
##  ARIMA(1,1,0)(0,1,0)[12]                    : -610.4453
##  ARIMA(1,1,0)(0,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,0)(1,1,0)[12]                    : -658.8733
##  ARIMA(1,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,0)[12]                    : -685.0633
##  ARIMA(1,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,2)[12]                    : -818.1563
##  ARIMA(1,1,1)(0,1,0)[12]                    : -816.3186
##  ARIMA(1,1,1)(0,1,1)[12]                    : -959.6487
##  ARIMA(1,1,1)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,1)(1,1,0)[12]                    : -863.3432
##  ARIMA(1,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,1)(1,1,2)[12]                    : Inf
##  ARIMA(1,1,1)(2,1,0)[12]                    : -890.1975
##  ARIMA(1,1,1)(2,1,1)[12]                    : -967.5769
##  ARIMA(1,1,2)(0,1,0)[12]                    : -820.9037
##  ARIMA(1,1,2)(0,1,1)[12]                    : -968.5302
##  ARIMA(1,1,2)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,2)(1,1,0)[12]                    : -870.1153
##  ARIMA(1,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,2)(2,1,0)[12]                    : -899.7004
##  ARIMA(1,1,3)(0,1,0)[12]                    : Inf
##  ARIMA(1,1,3)(0,1,1)[12]                    : -974.2582
##  ARIMA(1,1,3)(1,1,0)[12]                    : -880.4679
##  ARIMA(1,1,4)(0,1,0)[12]                    : Inf
##  ARIMA(2,1,0)(0,1,0)[12]                    : -716.0358
##  ARIMA(2,1,0)(0,1,1)[12]                    : -904.1772
##  ARIMA(2,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,0)[12]                    : -785.7711
##  ARIMA(2,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(2,1,0)[12]                    : -824.9017
##  ARIMA(2,1,0)(2,1,1)[12]                    : -906.4591
##  ARIMA(2,1,1)(0,1,0)[12]                    : -825.7094
##  ARIMA(2,1,1)(0,1,1)[12]                    : -970.19
##  ARIMA(2,1,1)(0,1,2)[12]                    : -974.8829
##  ARIMA(2,1,1)(1,1,0)[12]                    : -876.9427
##  ARIMA(2,1,1)(1,1,1)[12]                    : -973.7649
##  ARIMA(2,1,1)(2,1,0)[12]                    : -905.0924
##  ARIMA(2,1,2)(0,1,0)[12]                    : -828.5413
##  ARIMA(2,1,2)(0,1,1)[12]                    : -976.573
##  ARIMA(2,1,2)(1,1,0)[12]                    : -882.9126
##  ARIMA(2,1,3)(0,1,0)[12]                    : Inf
##  ARIMA(3,1,0)(0,1,0)[12]                    : -729.5044
##  ARIMA(3,1,0)(0,1,1)[12]                    : -907.4711
##  ARIMA(3,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(3,1,0)(1,1,0)[12]                    : -793.9063
##  ARIMA(3,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(3,1,0)(2,1,0)[12]                    : -828.2291
##  ARIMA(3,1,1)(0,1,0)[12]                    : -825.1369
##  ARIMA(3,1,1)(0,1,1)[12]                    : -971.827
##  ARIMA(3,1,1)(1,1,0)[12]                    : -878.57
##  ARIMA(3,1,2)(0,1,0)[12]                    : -826.5484
##  ARIMA(4,1,0)(0,1,0)[12]                    : -775.049
##  ARIMA(4,1,0)(0,1,1)[12]                    : -937.8025
##  ARIMA(4,1,0)(1,1,0)[12]                    : -833.1653
##  ARIMA(4,1,1)(0,1,0)[12]                    : -830.0257
##  ARIMA(5,1,0)(0,1,0)[12]                    : -800.1717
##
##
##
##  Best model: ARIMA(0,1,3)(0,1,2)[12]
```

```
auto_select_AIC
```

```
## Series: dat_short_log
## ARIMA(0,1,3)(0,1,2)[12]
##
## Coefficients:
##           ma1      ma2     ma3     sma1     sma2
##       -1.0663  -0.0187  0.1943  -0.7240  -0.1411
## s.e.   0.0534   0.0883  0.0627   0.0532   0.0520
##
## sigma^2 estimated as 0.004704:  log likelihood=494.31
## AIC=-976.61   AICc=-976.4   BIC=-952.71
```

```
auto_select_BIC = auto.arima(dat_short_log,stepwise=FALSE,seasonal=TRUE,ic="bic",trace=TRUE,approximati
```

```
##
##  ARIMA(0,1,0)(0,1,0)[12]                    : -485.8656
##  ARIMA(0,1,0)(0,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,0)[12]                    : -529.5958
##  ARIMA(0,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,0)[12]                    : -541.7106
##  ARIMA(0,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,2)[12]                    : -680.0987
##  ARIMA(0,1,1)(0,1,0)[12]                    : -800.1321
##  ARIMA(0,1,1)(0,1,1)[12]                    : -938.9675
##  ARIMA(0,1,1)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,0)[12]                    : -841.2863
##  ARIMA(0,1,1)(1,1,1)[12]                     : Inf
##  ARIMA(0,1,1)(1,1,2)[12]                     : Inf
##  ARIMA(0,1,1)(2,1,0)[12]                     : -858.7695
##  ARIMA(0,1,1)(2,1,1)[12]                     : -940.1251
##  ARIMA(0,1,1)(2,1,2)[12]                     : -942.7635
##  ARIMA(0,1,2)(0,1,0)[12]                     : -808.3335
##  ARIMA(0,1,2)(0,1,1)[12]                     : -948.3516
##  ARIMA(0,1,2)(0,1,2)[12]                     : Inf
##  ARIMA(0,1,2)(1,1,0)[12]                     : -852.8852
##  ARIMA(0,1,2)(1,1,1)[12]                     : Inf
##  ARIMA(0,1,2)(1,1,2)[12]                     : -944.1818
##  ARIMA(0,1,2)(2,1,0)[12]                     : -878.0768
##  ARIMA(0,1,2)(2,1,1)[12]                     : -947.4965
##  ARIMA(0,1,3)(0,1,0)[12]                     : -807.8482
##  ARIMA(0,1,3)(0,1,1)[12]                     : -951.6425
##  ARIMA(0,1,3)(0,1,2)[12]                     : -952.7069
##  ARIMA(0,1,3)(1,1,0)[12]                     : -854.5749
##  ARIMA(0,1,3)(1,1,1)[12]                     : -951.5326
##  ARIMA(0,1,3)(2,1,0)[12]                     : -878.6344
##  ARIMA(0,1,4)(0,1,0)[12]                     : -806.5539
##  ARIMA(0,1,4)(0,1,1)[12]                     : -947.1349
##  ARIMA(0,1,4)(1,1,0)[12]                     : -854.5285
##  ARIMA(0,1,5)(0,1,0)[12]                     : -801.6371
##  ARIMA(1,1,0)(0,1,0)[12]                     : -602.4774
##  ARIMA(1,1,0)(0,1,1)[12]                     : Inf
##  ARIMA(1,1,0)(0,1,2)[12]                     : Inf
##  ARIMA(1,1,0)(1,1,0)[12]                     : -646.9215
##  ARIMA(1,1,0)(1,1,1)[12]                     : Inf
##  ARIMA(1,1,0)(1,1,2)[12]                     : Inf
##  ARIMA(1,1,0)(2,1,0)[12]                     : -669.1276
##  ARIMA(1,1,0)(2,1,1)[12]                     : Inf
##  ARIMA(1,1,0)(2,1,2)[12]                     : -794.2527
##  ARIMA(1,1,1)(0,1,0)[12]                     : -804.3668
##  ARIMA(1,1,1)(0,1,1)[12]                     : -943.7129
##  ARIMA(1,1,1)(0,1,2)[12]                     : Inf
##  ARIMA(1,1,1)(1,1,0)[12]                     : -847.4075
##  ARIMA(1,1,1)(1,1,1)[12]                     : Inf
##  ARIMA(1,1,1)(1,1,2)[12]                     : Inf
##  ARIMA(1,1,1)(2,1,0)[12]                     : -870.2778
##  ARIMA(1,1,1)(2,1,1)[12]                     : -943.6732
##  ARIMA(1,1,2)(0,1,0)[12]                     : -804.9679
##  ARIMA(1,1,2)(0,1,1)[12]                     : -948.6106
##  ARIMA(1,1,2)(0,1,2)[12]                     : Inf
##  ARIMA(1,1,2)(1,1,0)[12]                     : -850.1956
##  ARIMA(1,1,2)(1,1,1)[12]                     : Inf
##  ARIMA(1,1,2)(2,1,0)[12]                     : -875.7968
##  ARIMA(1,1,3)(0,1,0)[12]                     : Inf
##  ARIMA(1,1,3)(0,1,1)[12]                     : -950.3546
##  ARIMA(1,1,3)(1,1,0)[12]                     : -856.5643
##  ARIMA(1,1,4)(0,1,0)[12]                     : Inf
##  ARIMA(2,1,0)(0,1,0)[12]                     : -704.084
##  ARIMA(2,1,0)(0,1,1)[12]                     : -888.2414
##  ARIMA(2,1,0)(0,1,2)[12]                     : Inf
##  ARIMA(2,1,0)(1,1,0)[12]                     : -769.8354
##  ARIMA(2,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(2,1,0)[12]                    : -804.9821
##  ARIMA(2,1,0)(2,1,1)[12]                    : -882.5555
##  ARIMA(2,1,1)(0,1,0)[12]                    : -809.7737
##  ARIMA(2,1,1)(0,1,1)[12]                    : -950.2703
##  ARIMA(2,1,1)(0,1,2)[12]                    : -950.9793
##  ARIMA(2,1,1)(1,1,0)[12]                    : -857.023
##  ARIMA(2,1,1)(1,1,1)[12]                    : -949.8613
##  ARIMA(2,1,1)(2,1,0)[12]                    : -881.1887
##  ARIMA(2,1,2)(0,1,0)[12]                    : -808.6216
##  ARIMA(2,1,2)(0,1,1)[12]                    : -952.6694
##  ARIMA(2,1,2)(1,1,0)[12]                    : -859.009
##  ARIMA(2,1,3)(0,1,0)[12]                    : Inf
##  ARIMA(3,1,0)(0,1,0)[12]                    : -713.5686
##  ARIMA(3,1,0)(0,1,1)[12]                    : -887.5514
##  ARIMA(3,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(3,1,0)(1,1,0)[12]                    : -773.9866
##  ARIMA(3,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(3,1,0)(2,1,0)[12]                    : -804.3255
##  ARIMA(3,1,1)(0,1,0)[12]                    : -805.2172
##  ARIMA(3,1,1)(0,1,1)[12]                    : -947.9233
##  ARIMA(3,1,1)(1,1,0)[12]                    : -854.6664
##  ARIMA(3,1,2)(0,1,0)[12]                    : -802.6448
##  ARIMA(4,1,0)(0,1,0)[12]                    : -755.1293
##  ARIMA(4,1,0)(0,1,1)[12]                    : -913.8989
##  ARIMA(4,1,0)(1,1,0)[12]                    : -809.2616
##  ARIMA(4,1,1)(0,1,0)[12]                    : -806.122
##  ARIMA(5,1,0)(0,1,0)[12]                    : -776.2681
## 
## 
## 
##  Best model: ARIMA(0,1,3)(0,1,2)[12]
```

```
auto_select_BIC
```

```
## Series: dat_short_log 
## ARIMA(0,1,3)(0,1,2)[12] 
## 
## Coefficients:
##           ma1      ma2     ma3     sma1     sma2
##       -1.0663  -0.0187  0.1943  -0.7240  -0.1411
## s.e.   0.0534   0.0883  0.0627   0.0532   0.0520
## 
## sigma^2 estimated as 0.004704:  log likelihood=494.31
## AIC=-976.61   AICc=-976.4   BIC=-952.71
```

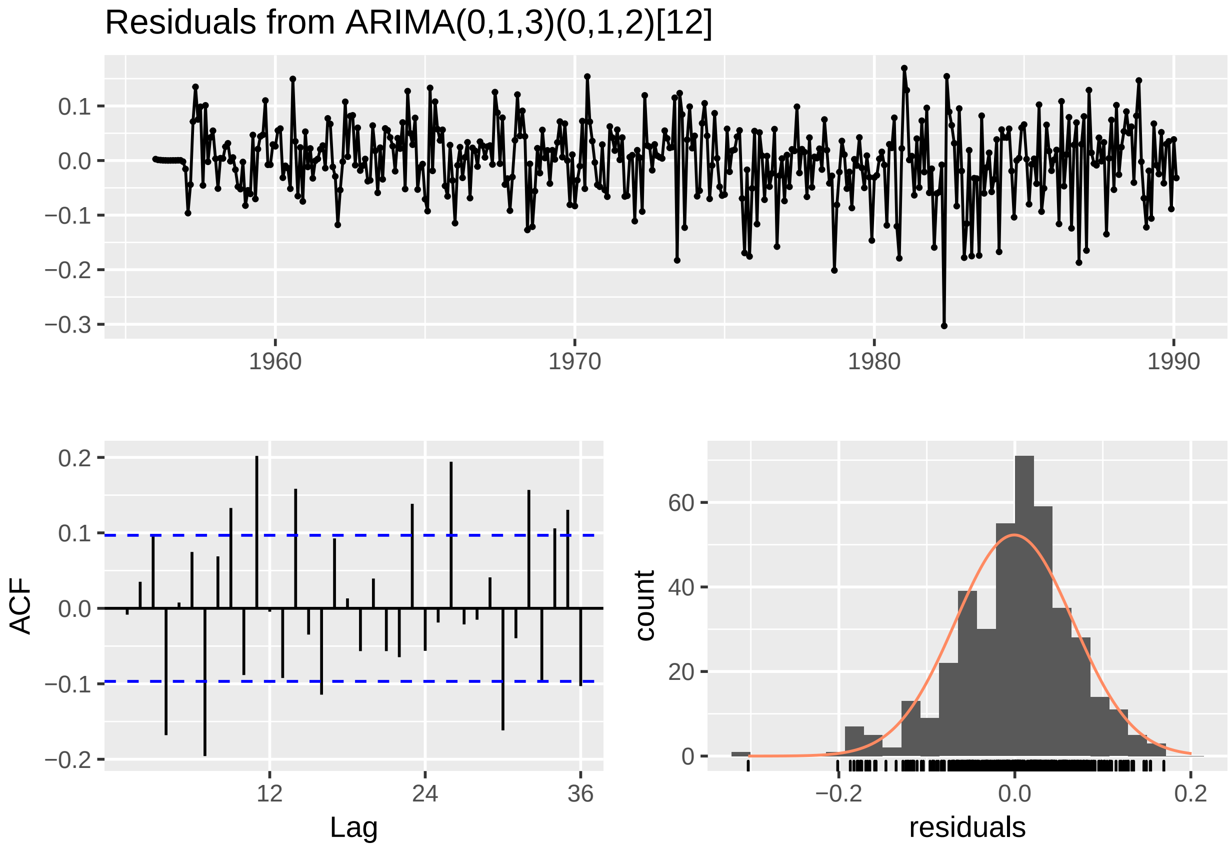
```
auto_select_AICC = auto.arima(dat_short_log,stepwise=FALSE,seasonal=TRUE,ic="aicc",trace=TRUE,approxima
```

```
## 
##  ARIMA(0,1,0)(0,1,0)[12]                    : -489.8394
##  ARIMA(0,1,0)(0,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,0)[12]                    : -537.5332
##  ARIMA(0,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,0)[12]                    : -553.6014
##  ARIMA(0,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,2)[12]                    : -699.865
##  ARIMA(0,1,1)(0,1,0)[12]                    : -808.0695
##  ARIMA(0,1,1)(0,1,1)[12]                    : -950.8583
##  ARIMA(0,1,1)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,0)[12]                    : -853.177
##  ARIMA(0,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(2,1,0)[12]                    : -874.6032
##  ARIMA(0,1,1)(2,1,1)[12]                    : -959.8914
##  ARIMA(0,1,1)(2,1,2)[12]                    : -966.4517
##  ARIMA(0,1,2)(0,1,0)[12]                    : -820.2242
##  ARIMA(0,1,2)(0,1,1)[12]                    : -964.1853
##  ARIMA(0,1,2)(0,1,2)[12]                    : Inf
##  ARIMA(0,1,2)(1,1,0)[12]                    : -868.719
##  ARIMA(0,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,2)(1,1,2)[12]                    : -967.87
##  ARIMA(0,1,2)(2,1,0)[12]                    : -897.843
##  ARIMA(0,1,2)(2,1,1)[12]                    : -971.1847
##  ARIMA(0,1,3)(0,1,0)[12]                    : -823.6819
##  ARIMA(0,1,3)(0,1,1)[12]                    : -971.4088
##  ARIMA(0,1,3)(0,1,2)[12]                    : -976.3952
##  ARIMA(0,1,3)(1,1,0)[12]                    : -874.3412
##  ARIMA(0,1,3)(1,1,1)[12]                    : -975.2208
##  ARIMA(0,1,3)(2,1,0)[12]                    : -902.3227
##  ARIMA(0,1,4)(0,1,0)[12]                    : -826.3201
##  ARIMA(0,1,4)(0,1,1)[12]                    : -970.8231
##  ARIMA(0,1,4)(1,1,0)[12]                    : -878.2167
##  ARIMA(0,1,5)(0,1,0)[12]                    : -825.3253
##  ARIMA(1,1,0)(0,1,0)[12]                    : -610.4148
##  ARIMA(1,1,0)(0,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,0)(1,1,0)[12]                    : -658.8122
##  ARIMA(1,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,0)[12]                    : -684.9613
##  ARIMA(1,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,2)[12]                    : -817.9409
##  ARIMA(1,1,1)(0,1,0)[12]                    : -816.2575
##  ARIMA(1,1,1)(0,1,1)[12]                    : -959.5467
##  ARIMA(1,1,1)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,1)(1,1,0)[12]                    : -863.2412
##  ARIMA(1,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,1)(1,1,2)[12]                    : Inf
##  ARIMA(1,1,1)(2,1,0)[12]                    : -890.044
##  ARIMA(1,1,1)(2,1,1)[12]                    : -967.3615
##  ARIMA(1,1,2)(0,1,0)[12]                    : -820.8016
##  ARIMA(1,1,2)(0,1,1)[12]                    : -968.3768
##  ARIMA(1,1,2)(0,1,2)[12]                    : Inf
##  ARIMA(1,1,2)(1,1,0)[12]                    : -869.9618
##  ARIMA(1,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,2)(2,1,0)[12]                    : -899.4851
##  ARIMA(1,1,3)(0,1,0)[12]                    : Inf
##  ARIMA(1,1,3)(0,1,1)[12]                    : -974.0429
##  ARIMA(1,1,3)(1,1,0)[12]                    : -880.2525
##  ARIMA(1,1,4)(0,1,0)[12]                    : Inf
##  ARIMA(2,1,0)(0,1,0)[12]                    : -715.9747
##  ARIMA(2,1,0)(0,1,1)[12]                    : -904.0751
##  ARIMA(2,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,0)[12]                    : -785.6691
##  ARIMA(2,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(2,1,0)[12]                    : -824.7483
##  ARIMA(2,1,0)(2,1,1)[12]                    : -906.2437
##  ARIMA(2,1,1)(0,1,0)[12]                    : -825.6074
##  ARIMA(2,1,1)(0,1,1)[12]                    : -970.0365
##  ARIMA(2,1,1)(0,1,2)[12]                    : -974.6675
##  ARIMA(2,1,1)(1,1,0)[12]                    : -876.7893
##  ARIMA(2,1,1)(1,1,1)[12]                    : -973.5495
##  ARIMA(2,1,1)(2,1,0)[12]                    : -904.877
##  ARIMA(2,1,2)(0,1,0)[12]                    : -828.3878
##  ARIMA(2,1,2)(0,1,1)[12]                    : -976.3577
##  ARIMA(2,1,2)(1,1,0)[12]                    : -882.6972
##  ARIMA(2,1,3)(0,1,0)[12]                    : Inf
##  ARIMA(3,1,0)(0,1,0)[12]                    : -729.4023
##  ARIMA(3,1,0)(0,1,1)[12]                    : -907.3176
##  ARIMA(3,1,0)(0,1,2)[12]                    : Inf
##  ARIMA(3,1,0)(1,1,0)[12]                    : -793.7528
##  ARIMA(3,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(3,1,0)(2,1,0)[12]                    : -828.0137
##  ARIMA(3,1,1)(0,1,0)[12]                    : -824.9834
##  ARIMA(3,1,1)(0,1,1)[12]                    : -971.6116
##  ARIMA(3,1,1)(1,1,0)[12]                    : -878.3546
##  ARIMA(3,1,2)(0,1,0)[12]                    : -826.333
##  ARIMA(4,1,0)(0,1,0)[12]                    : -774.8956
##  ARIMA(4,1,0)(0,1,1)[12]                    : -937.5871
##  ARIMA(4,1,0)(1,1,0)[12]                    : -832.9499
##  ARIMA(4,1,1)(0,1,0)[12]                    : -829.8103
##  ARIMA(5,1,0)(0,1,0)[12]                    : -799.9563
##
##
##
##  Best model: ARIMA(0,1,3)(0,1,2)[12]
```

```
auto_select_AICC
```

```
## Series: dat_short_log
## ARIMA(0,1,3)(0,1,2)[12]
##
## Coefficients:
##           ma1      ma2     ma3     sma1     sma2
##       -1.0663  -0.0187  0.1943  -0.7240  -0.1411
## s.e.   0.0534   0.0883  0.0627   0.0532   0.0520
##
## sigma^2 estimated as 0.004704:  log likelihood=494.31
## AIC=-976.61   AICc=-976.4   BIC=-952.71
```

```
checkresiduals(auto_select_AIC)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,3)(0,1,2)[12]
## Q* = 103.92, df = 19, p-value = 1.038e-13
##
## Model df: 5.   Total lags used: 24
```

Thus the best model is a seasonal ARIMA model as shown.

b)

The confidence intervals will be given by:

```
confint(auto_select_AIC, level=0.95)
```

```
##             2.5 %      97.5 %
## ma1  -1.17093503 -0.96169878
## ma2  -0.19178794  0.15441118
## ma3   0.07144952  0.31709330
## sma1 -0.82828910 -0.61962405
## sma2 -0.24306446 -0.03920016
```

c)

The ACF doesn't look too good in the sense that there is significant correlation still present; with many spikes outside the confidence bounds. Let's run some tests:

```
adf.test(residuals(auto_select_AIC))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  residuals(auto_select_AIC)
## Dickey-Fuller = -8.0736, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

The null hypothesis non-starionarity can be rejected for upto 99% confidence as we can see from the p value from the ADF test; and for KPSS:

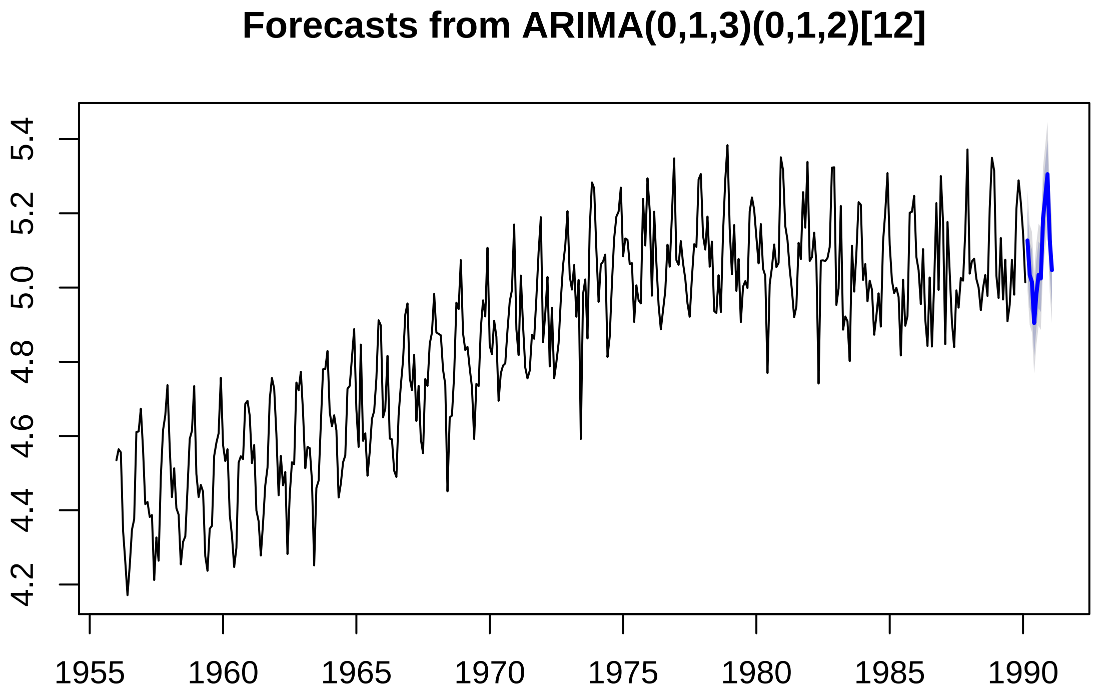
```
kpss.test(residuals(auto_select_AIC))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  residuals(auto_select_AIC)
## KPSS Level = 0.61499, Truncation lag parameter = 4, p-value =
## 0.02127
```

Starionarity is null here, but it is still able to be rejected with high confidence as seen from the p value. The tests don't agree; and with the ACF graph as well, it is unlikely that the residual is stationary, thus it's not likely white noise.

d)

```
fcast <- forecast((auto_select_AIC), h=12)
plot(fcast)
```



e)

Thus the predicted next 12 values and the corresponding 95% confidence bounds can be seen from the following output:

```
summary(fcast)
```

```
##
## Forecast method: ARIMA(0,1,3)(0,1,2)[12]
##
## Model Information:
## Series: dat_short_log
## ARIMA(0,1,3)(0,1,2)[12]
##
## Coefficients:
##           ma1      ma2     ma3     sma1     sma2
##       -1.0663  -0.0187  0.1943  -0.7240  -0.1411
## s.e.   0.0534   0.0883  0.0627   0.0532   0.0520
##
## sigma^2 estimated as 0.004704:  log likelihood=494.31
## AIC=-976.61   AICc=-976.4   BIC=-952.71
##
## Error measures:
##                         ME       RMSE        MAE         MPE     MAPE
## Training set -0.0005984734 0.06706188 0.05045498 -0.01413405 1.036919
##                   MASE         ACF1
## Training set 0.7303307 -0.008357447
```

```
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 1990       5.127031 5.039135 5.214928 4.992605 5.261457
## Apr 1990       5.034187 4.946097 5.122276 4.899465 5.168908
## May 1990       5.014801 4.926395 5.103207 4.879596 5.150006
## Jun 1990       4.904277 4.815351 4.993203 4.768277 5.040278
## Jul 1990       4.982070 4.892627 5.071513 4.845279 5.118861
## Aug 1990       5.034250 4.944293 5.124208 4.896673 5.171828
## Sep 1990       5.024294 4.933826 5.114763 4.885935 5.162654
## Oct 1990       5.186463 5.095486 5.277439 5.047326 5.325599
## Nov 1990       5.243338 5.151856 5.334821 5.103428 5.383248
## Dec 1990       5.305418 5.213433 5.397403 5.164740 5.446097
## Jan 1991       5.130140 5.037655 5.222625 4.988697 5.271583
## Feb 1991       5.046877 4.953895 5.139859 4.904673 5.189081
```

Where "Point Forecast" corresponds to the predicted values, and "Lo 95" and "Hi 95" are the corresponding lower and upper bounds, respectively, for the 95% prediction bounds.

f)

Below is a comparison of the (logged) actual values vs the forecasted values:

```
i<-1:12
beer_dat_log <- log(beer_dat)
vals <-(as.numeric(tail(beer_dat_log,12)))
predicted<-(as.numeric(fcast$mean))
cbind(vals,predicted)
```

```
##           vals predicted
##  [1,] 5.093750  5.127031
##  [2,] 5.034352  5.034187
##  [3,] 5.029130  5.014801
##  [4,] 4.908233  4.904277
##  [5,] 4.999237  4.982070
##  [6,] 4.999237  5.034250
##  [7,] 4.894101  5.024294
##  [8,] 5.266827  5.186463
##  [9,] 5.339459  5.243338
## [10,] 5.283204  5.305418
## [11,] 5.099866  5.130140
## [12,] 4.997212  5.046877
```

Now we display the errors:

```
errors<-abs(vals-predicted)
cbind(i,errors)
```

```
##        i       errors
##  [1,]  1 0.0332810075
##  [2,]  2 0.0001652292
##  [3,]  3 0.0143288433
##  [4,]  4 0.0039560328
##  [5,]  5 0.0171674452
##  [6,]  6 0.0350132317
##  [7,]  7 0.1301927777
##  [8,]  8 0.0803641450
##  [9,]  9 0.0961208594
## [10,] 10 0.0222145871
## [11,] 11 0.0302733733
## [12,] 12 0.0496648714
```

Let's see what the last (logged) value of the original series is:

```
last_value <- tail(beer_dat_log,1)
last_value
```

```
##           Feb
## 1991 4.997212
```

4.997212 is indeed within the prediction bounds for the last value: [4.904673,5.189081].

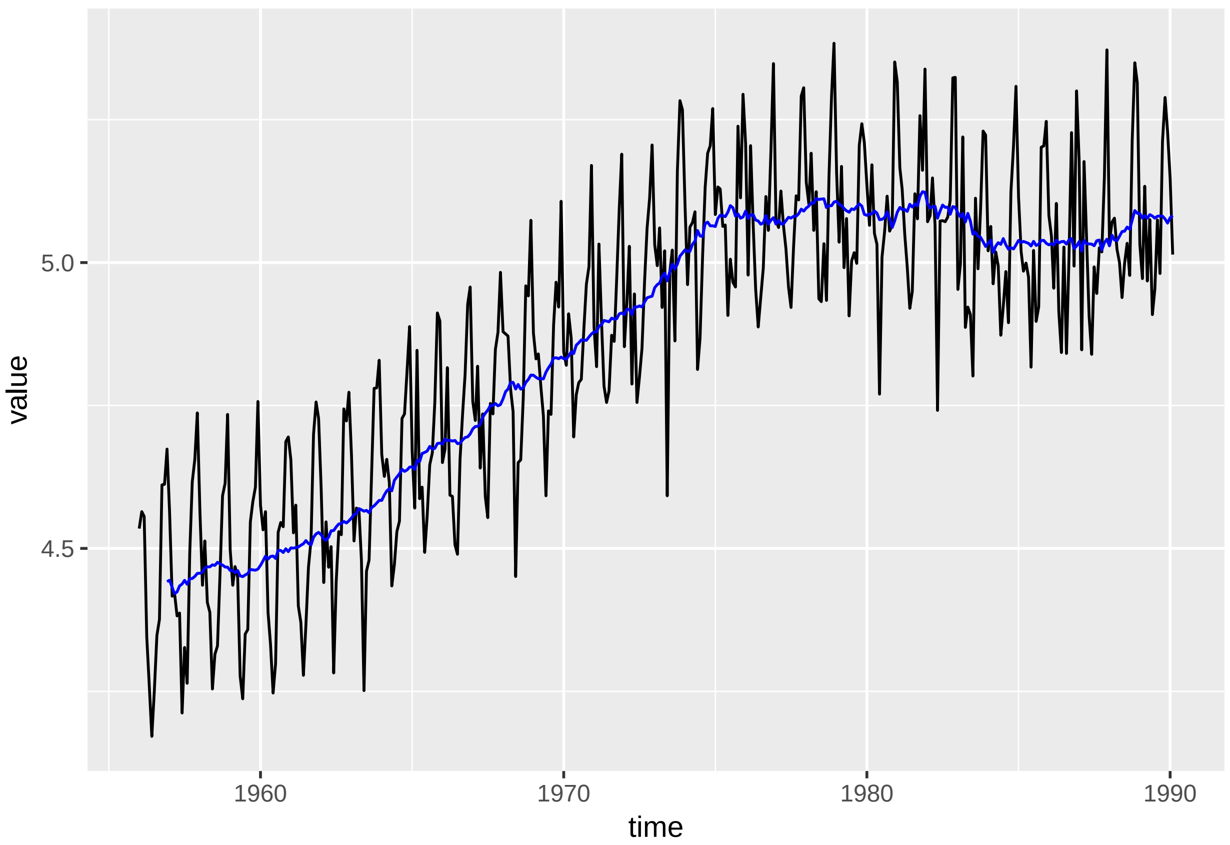
6.10)

We shall now take the classical decomposition approach. Below we establish our objects and make sure they're indexed properly, and compute an initial trend with SMA 12:

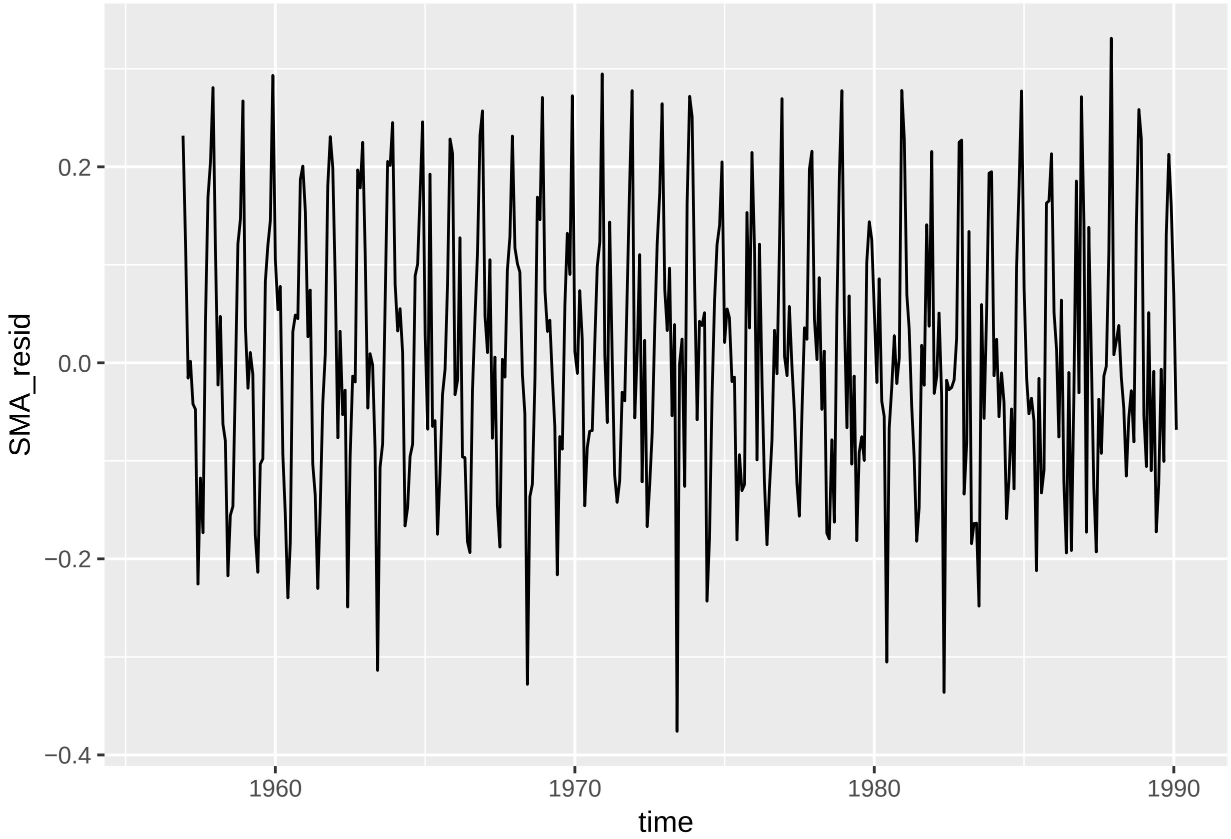
```
dat_df = ts_df(dat_short_log) %>% rename(time = time,
                                             value = value)
dat_tbl = as_tbl_time(dat_df,index=time)

trend_comp = ts_df(SMA(dat_short_log,n=12)) %>% rename(time=time,SMA_12=value)


dat_tbl  =  full_join(dat_tbl,trend_comp) %>%
                    mutate(SMA_resid=dat_short_log-SMA_12)
```

Now we display the initial trend:

```
ggplot(dat_tbl,aes(x=time,y=value)) + geom_line() +
  geom_line(aes(y=SMA_12),color="blue")
```



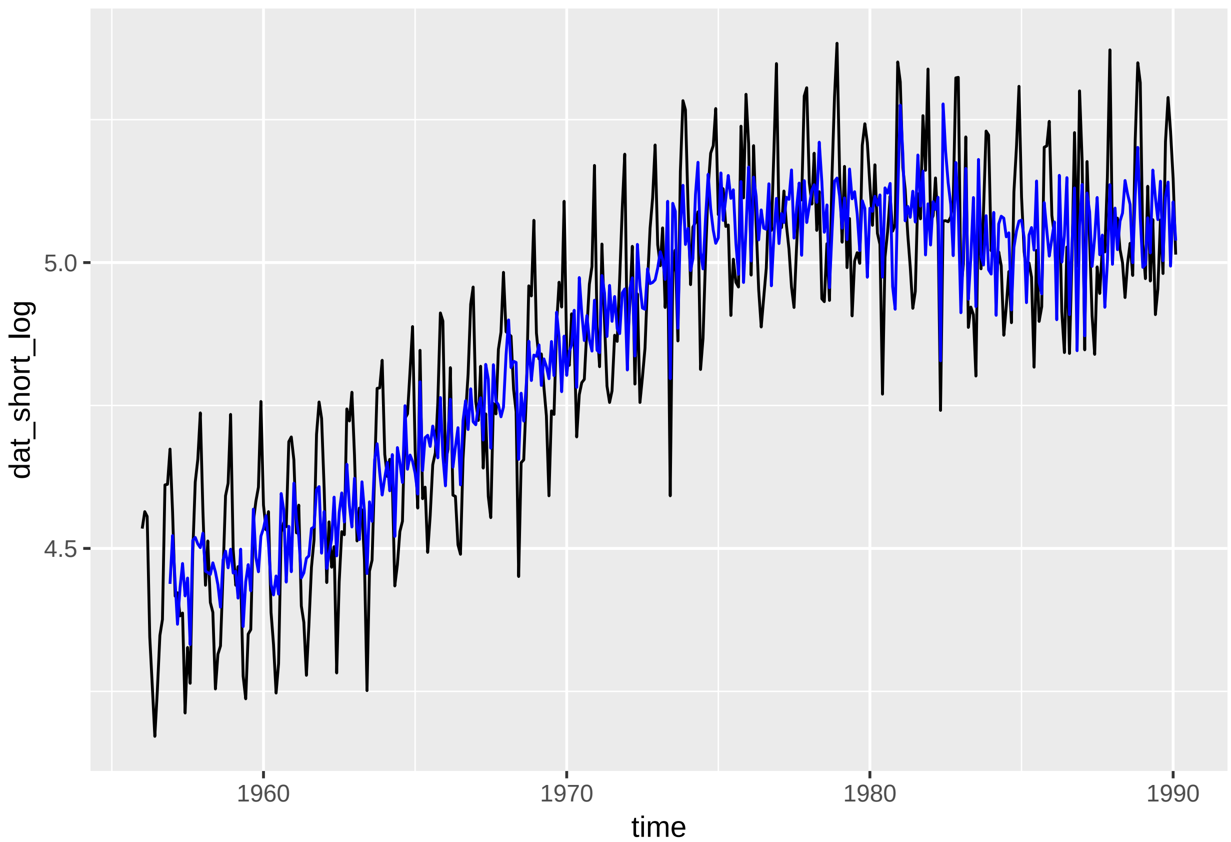
```
ggplot(dat_tbl,aes(x=time,y=SMA_resid)) + geom_line()
```



As we can see the trend has fallen off. Now we will handle seasonality:

```
SMA_resid_ts = ts(dat_tbl%>% filter(!is.na(SMA_resid)) %>% pull(SMA_resid),
   start=c(1956,12),frequency=12)

season_comp = season(SMA_resid_ts,d=12)

SMA_resid_tbl = ts_df(SMA_resid_ts) %>% rename(time=time,
                                               SMA_resid=value) %>%
                  mutate(seasonal = season_comp) %>% as_tbl_time(index=time)

dat_tbl = full_join(dat_tbl,SMA_resid_tbl)
```

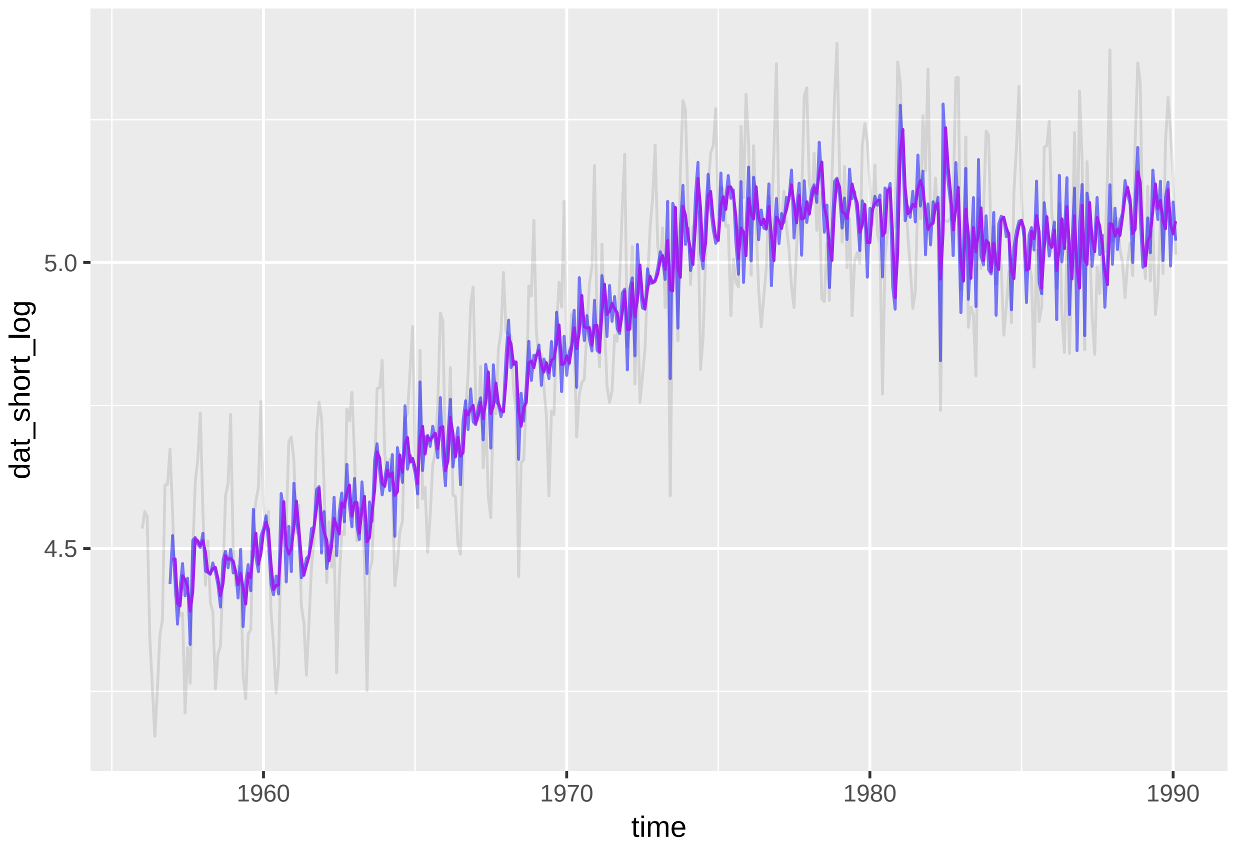
```
dat_tbl = dat_tbl %>% mutate(deseason=dat_short_log-seasonal)


ggplot(dat_tbl,aes(x=time,y=dat_short_log)) + geom_line() +
  geom_line(aes(y=deseason),color="blue")
```



Finally we recompute a quadriatic trend and and plot it together with the seasonal component:

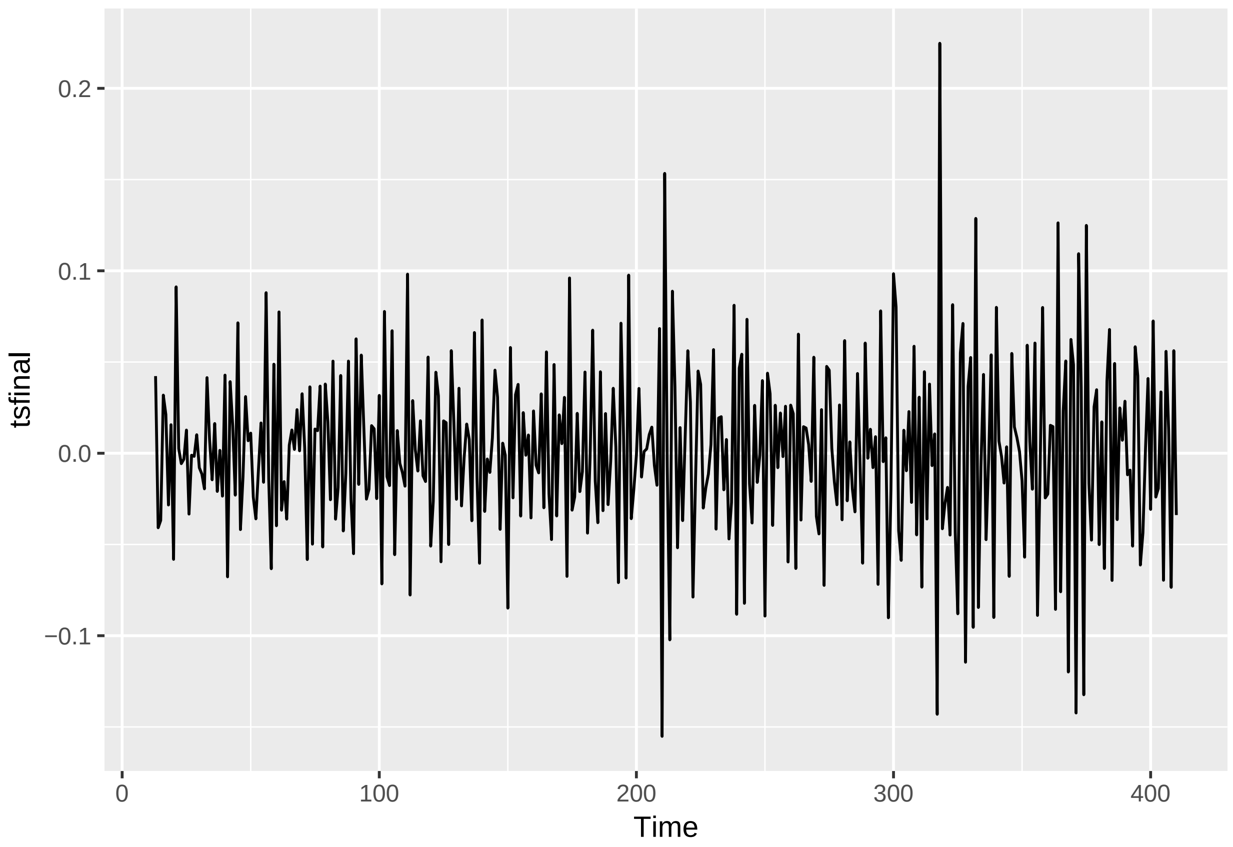
```
deseason_ts = ts(dat_tbl %>% filter(!is.na(deseason)) %>% pull(deseason),
                 start=c(1956,12),frequency=12)

deseason_trend = ts_df(SMA(deseason_ts,n=2)) %>% rename(time=time,de_SMA_2 = value)

dat_tbl = full_join(dat_tbl,deseason_trend)
```

```
dat_tbl = dat_tbl %>% mutate(Final_resid=dat_short_log-de_SMA_2-seasonal)

ggplot(dat_tbl,aes(x=time,y=dat_short_log)) +geom_line(alpha=0.1) + geom_line(aes(y=deseason),color="blu
```

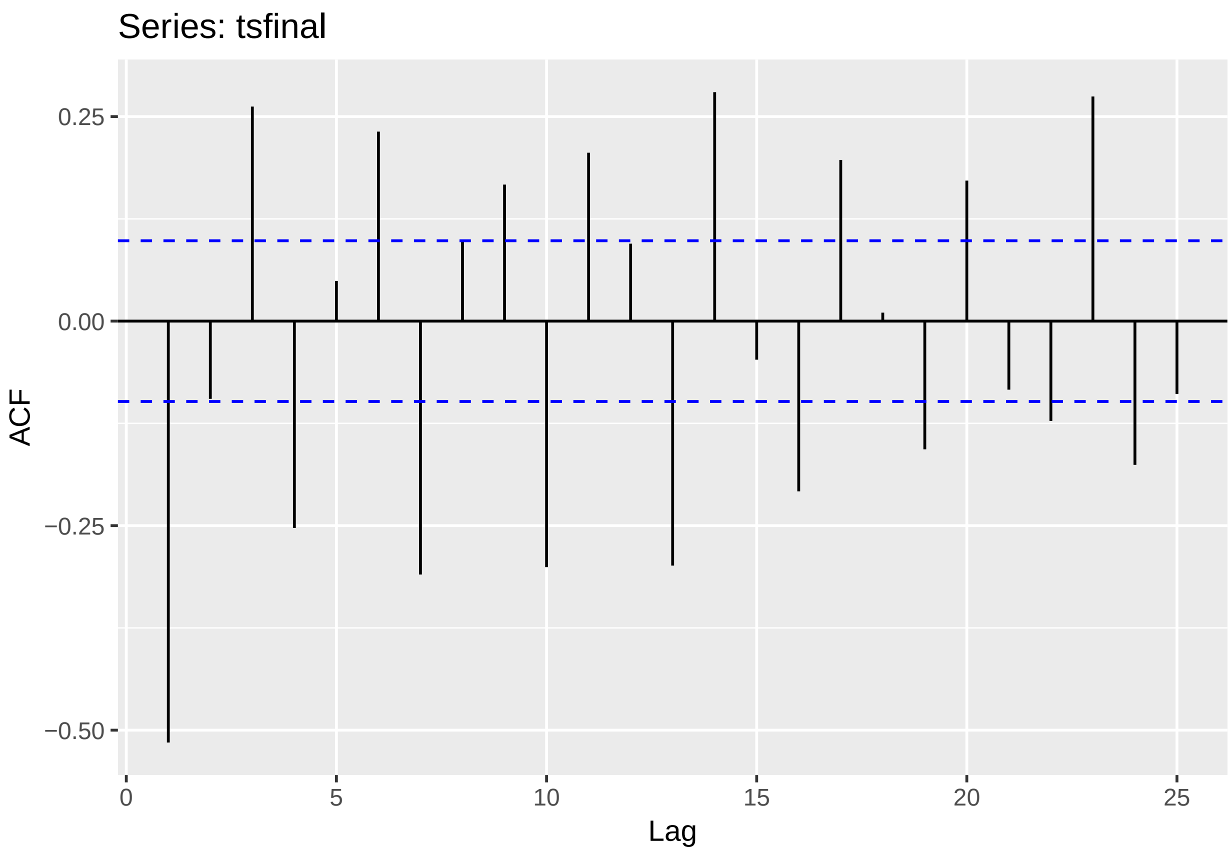


The final residuals post decomposition look like this:

```
t_s = ts(dat_tbl %>% pull(Final_resid))

tsfinal = na.remove(t_s)
autoplot(tsfinal)
```



```
ggAcf(tsfinal)
```



The new ACF is significantly less patterned than the initial ACF, but doesn't look stationary; there is significant correlation remaining.

a)

Now we find the ARIMA model of best fit for the residuals:

```
auto_select_AIC2 = auto.arima(tsfinal ,stepwise=FALSE,seasonal=FALSE,ic="aic",trace=TRUE,approximation=
```

```
##
##  ARIMA(0,0,0) with zero mean     : -1278.225
##  ARIMA(0,0,0) with non-zero mean : -1276.322
##  ARIMA(0,0,1) with zero mean     : -1541.588
##  ARIMA(0,0,1) with non-zero mean : -1551.519
##  ARIMA(0,0,2) with zero mean     : -1550.957
##  ARIMA(0,0,2) with non-zero mean : -1560.68
##  ARIMA(0,0,3) with zero mean     : -1556.101
##  ARIMA(0,0,3) with non-zero mean : -1565.694
##  ARIMA(0,0,4) with zero mean     : -1555.667
##  ARIMA(0,0,4) with non-zero mean : -1565.183
##  ARIMA(0,0,5) with zero mean     : -1575.871
##  ARIMA(0,0,5) with non-zero mean : -1583.606
##  ARIMA(1,0,0) with zero mean     : -1398.884
##  ARIMA(1,0,0) with non-zero mean : -1397.181
##  ARIMA(1,0,1) with zero mean     : -1547.922
##  ARIMA(1,0,1) with non-zero mean : -1557.799
##  ARIMA(1,0,2) with zero mean     : -1571.45
##  ARIMA(1,0,2) with non-zero mean : -1575.054
##  ARIMA(1,0,3) with zero mean     : -1558.368
##  ARIMA(1,0,3) with non-zero mean : -1567.93
##  ARIMA(1,0,4) with zero mean     : -1556.499
##  ARIMA(1,0,4) with non-zero mean : -1566.059
##  ARIMA(2,0,0) with zero mean     : -1505.514
##  ARIMA(2,0,0) with non-zero mean : -1504.352
##  ARIMA(2,0,1) with zero mean     : -1555.44
##  ARIMA(2,0,1) with non-zero mean : -1564.884
##  ARIMA(2,0,2) with zero mean     : -1561.124
##  ARIMA(2,0,2) with non-zero mean : -1570.582
##  ARIMA(2,0,3) with zero mean     : Inf
##  ARIMA(2,0,3) with non-zero mean : Inf
##  ARIMA(3,0,0) with zero mean     : -1508.278
##  ARIMA(3,0,0) with non-zero mean : -1507.352
##  ARIMA(3,0,1) with zero mean     : -1557.306
##  ARIMA(3,0,1) with non-zero mean : -1566.675
##  ARIMA(3,0,2) with zero mean     : -1560.47
##  ARIMA(3,0,2) with non-zero mean : -1569.959
##  ARIMA(4,0,0) with zero mean     : -1536.275
##  ARIMA(4,0,0) with non-zero mean : -1536.28
##  ARIMA(4,0,1) with zero mean     : -1571.411
##  ARIMA(4,0,1) with non-zero mean : -1581.218
##  ARIMA(5,0,0) with zero mean     : -1567.313
##  ARIMA(5,0,0) with non-zero mean : -1568.997
##
##
##
##  Best model: ARIMA(0,0,5) with non-zero mean
```

```
auto_select_AIC2
```

```
## Series: tsfinal
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##           ma1     ma2     ma3      ma4     ma5   mean
##       -1.0525  0.0381  0.1146  -0.2019  0.2492  8e-04
## s.e.   0.0488  0.0682  0.0698   0.0602  0.0499  2e-04
##
## sigma^2 estimated as 0.001067:  log likelihood=798.8
## AIC=-1583.61   AICc=-1583.32   BIC=-1555.7
```

```
auto_select_BIC2 = auto.arima(tsfinal,stepwise=FALSE,seasonal=FALSE,ic="bic",trace=TRUE,approximation=F
```

```
##
##  ARIMA(0,0,0) with zero mean     : -1274.239
##  ARIMA(0,0,0) with non-zero mean : -1268.349
##  ARIMA(0,0,1) with zero mean     : -1533.615
##  ARIMA(0,0,1) with non-zero mean : -1539.559
##  ARIMA(0,0,2) with zero mean     : -1538.998
##  ARIMA(0,0,2) with non-zero mean : -1544.735
##  ARIMA(0,0,3) with zero mean     : -1540.155
##  ARIMA(0,0,3) with non-zero mean : -1545.762
##  ARIMA(0,0,4) with zero mean     : -1535.735
##  ARIMA(0,0,4) with non-zero mean : -1541.265
##  ARIMA(0,0,5) with zero mean     : -1551.953
##  ARIMA(0,0,5) with non-zero mean : -1555.701
##  ARIMA(1,0,0) with zero mean     : -1390.911
##  ARIMA(1,0,0) with non-zero mean : -1385.221
##  ARIMA(1,0,1) with zero mean     : -1535.963
##  ARIMA(1,0,1) with non-zero mean : -1541.853
##  ARIMA(1,0,2) with zero mean     : -1555.504
##  ARIMA(1,0,2) with non-zero mean : -1555.121
##  ARIMA(1,0,3) with zero mean     : -1538.435
##  ARIMA(1,0,3) with non-zero mean : -1544.012
##  ARIMA(1,0,4) with zero mean     : -1532.58
##  ARIMA(1,0,4) with non-zero mean : -1538.153
##  ARIMA(2,0,0) with zero mean     : -1493.555
##  ARIMA(2,0,0) with non-zero mean : -1488.406
##  ARIMA(2,0,1) with zero mean     : -1539.494
##  ARIMA(2,0,1) with non-zero mean : -1544.952
##  ARIMA(2,0,2) with zero mean     : -1541.191
##  ARIMA(2,0,2) with non-zero mean : -1546.664
##  ARIMA(2,0,3) with zero mean     : Inf
##  ARIMA(2,0,3) with non-zero mean : Inf
##  ARIMA(3,0,0) with zero mean     : -1492.332
##  ARIMA(3,0,0) with non-zero mean : -1487.419
##  ARIMA(3,0,1) with zero mean     : -1537.374
##  ARIMA(3,0,1) with non-zero mean : -1542.757
##  ARIMA(3,0,2) with zero mean     : -1536.551
##  ARIMA(3,0,2) with non-zero mean : -1542.054
##  ARIMA(4,0,0) with zero mean     : -1516.343
##  ARIMA(4,0,0) with non-zero mean : -1512.361
##  ARIMA(4,0,1) with zero mean     : -1547.492
##  ARIMA(4,0,1) with non-zero mean : -1553.313
##  ARIMA(5,0,0) with zero mean     : -1543.394
##  ARIMA(5,0,0) with non-zero mean : -1541.092
##
##
##
##  Best model: ARIMA(0,0,5) with non-zero mean
```

```
auto_select_BIC2
```

```
## Series: tsfinal
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##           ma1     ma2     ma3      ma4     ma5   mean
##       -1.0525  0.0381  0.1146  -0.2019  0.2492  8e-04
## s.e.   0.0488  0.0682  0.0698   0.0602  0.0499  2e-04
##
## sigma^2 estimated as 0.001067:  log likelihood=798.8
## AIC=-1583.61   AICc=-1583.32   BIC=-1555.7
```

```
auto_select_AICC2 = auto.arima(tsfinal ,stepwise=FALSE,seasonal=FALSE,ic="aicc",trace=TRUE,approximatio
```

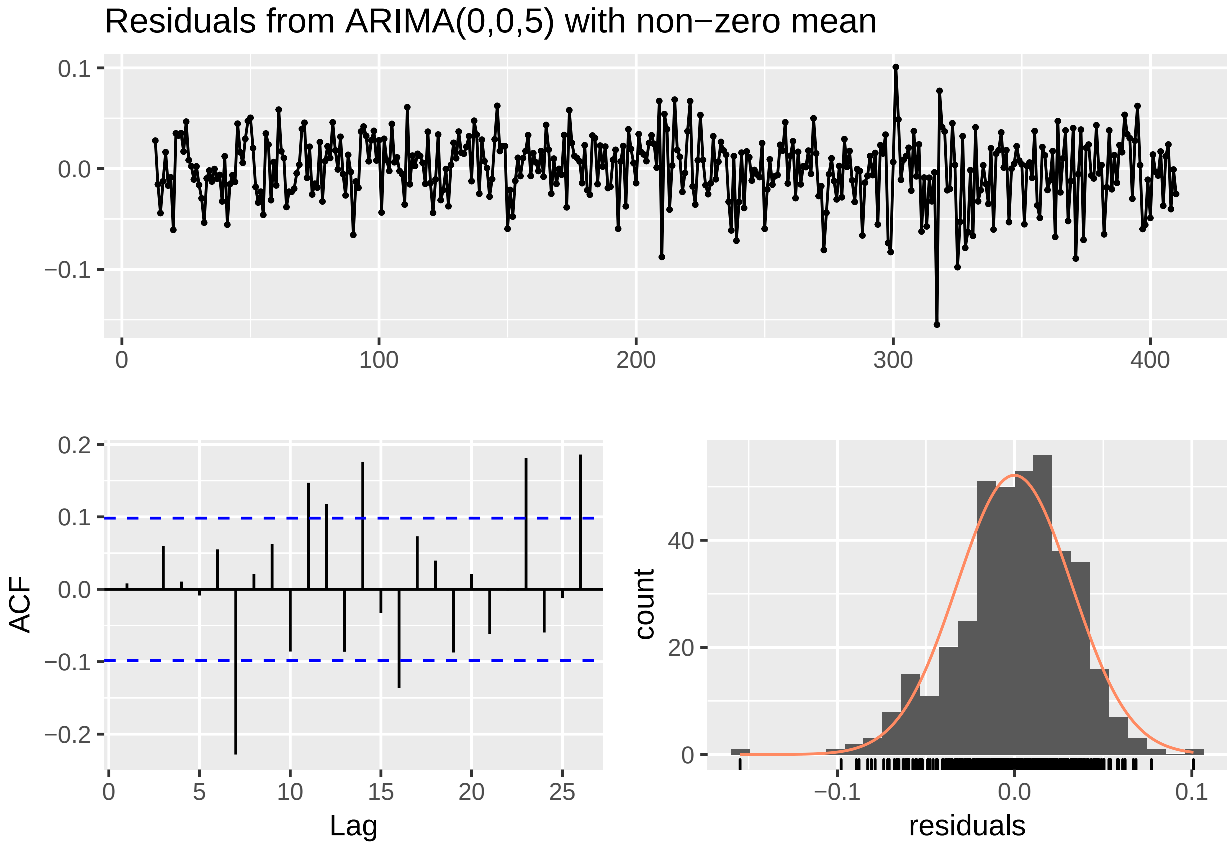
```
##
##  ARIMA(0,0,0) with zero mean     : -1278.215
##  ARIMA(0,0,0) with non-zero mean : -1276.291
##  ARIMA(0,0,1) with zero mean     : -1541.558
##  ARIMA(0,0,1) with non-zero mean : -1551.458
##  ARIMA(0,0,2) with zero mean     : -1550.896
##  ARIMA(0,0,2) with non-zero mean : -1560.579
##  ARIMA(0,0,3) with zero mean     : -1555.999
##  ARIMA(0,0,3) with non-zero mean : -1565.541
##  ARIMA(0,0,4) with zero mean     : -1555.514
##  ARIMA(0,0,4) with non-zero mean : -1564.968
##  ARIMA(0,0,5) with zero mean     : -1575.657
##  ARIMA(0,0,5) with non-zero mean : -1583.319
##  ARIMA(1,0,0) with zero mean     : -1398.854
##  ARIMA(1,0,0) with non-zero mean : -1397.12
##  ARIMA(1,0,1) with zero mean     : -1547.861
##  ARIMA(1,0,1) with non-zero mean : -1557.697
##  ARIMA(1,0,2) with zero mean     : -1571.348
##  ARIMA(1,0,2) with non-zero mean : -1574.901
##  ARIMA(1,0,3) with zero mean     : -1558.215
##  ARIMA(1,0,3) with non-zero mean : -1567.716
##  ARIMA(1,0,4) with zero mean     : -1556.284
##  ARIMA(1,0,4) with non-zero mean : -1565.771
##  ARIMA(2,0,0) with zero mean     : -1505.453
##  ARIMA(2,0,0) with non-zero mean : -1504.25
##  ARIMA(2,0,1) with zero mean     : -1555.338
##  ARIMA(2,0,1) with non-zero mean : -1564.731
##  ARIMA(2,0,2) with zero mean     : -1560.971
##  ARIMA(2,0,2) with non-zero mean : -1570.368
##  ARIMA(2,0,3) with zero mean     : Inf
##  ARIMA(2,0,3) with non-zero mean : Inf
##  ARIMA(3,0,0) with zero mean     : -1508.176
##  ARIMA(3,0,0) with non-zero mean : -1507.198
##  ARIMA(3,0,1) with zero mean     : -1557.153
##  ARIMA(3,0,1) with non-zero mean : -1566.461
##  ARIMA(3,0,2) with zero mean     : -1560.255
##  ARIMA(3,0,2) with non-zero mean : -1569.672
##  ARIMA(4,0,0) with zero mean     : -1536.122
##  ARIMA(4,0,0) with non-zero mean : -1536.065
##  ARIMA(4,0,1) with zero mean     : -1571.196
##  ARIMA(4,0,1) with non-zero mean : -1580.931
##  ARIMA(5,0,0) with zero mean     : -1567.098
##  ARIMA(5,0,0) with non-zero mean : -1568.71
##
##
##
##  Best model: ARIMA(0,0,5) with non-zero mean
```

```
auto_select_AICC2
```

```
## Series: tsfinal
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##            ma1     ma2     ma3      ma4     ma5   mean
##        -1.0525  0.0381  0.1146  -0.2019  0.2492  8e-04
## s.e.    0.0488  0.0682  0.0698   0.0602  0.0499  2e-04
```

```
##
## sigma^2 estimated as 0.001067:  log likelihood=798.8
## AIC=-1583.61   AICc=-1583.32   BIC=-1555.7
```

```
checkresiduals(auto_select_AIC2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,5) with non-zero mean
## Q* = 28.763, df = 4, p-value = 8.735e-06
##
## Model df: 6.   Total lags used: 10
```

Thus it would seem that an MA(5) model with non-zero mean fits best for the remaining residuals. It is worth noting that this is not the same model we got when we applied differencing instead.

b)

The confience intervals will be given by:

```
confint(auto_select_AIC2, level=0.95)
```

```
##                     2.5 %       97.5 %
## ma1       -1.1482391193 -0.956797027
## ma2       -0.0954728643  0.171676101
## ma3       -0.0222622769  0.251407281
## ma4       -0.3198462281 -0.083896425
## ma5        0.1514461665  0.346872271
## intercept  0.0003267463  0.001289422
```

c)

The ACF is more "noisy" but as there are significant correlation spikes present.

```
adf.test(residuals(auto_select_AIC2))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  residuals(auto_select_AIC2)
## Dickey-Fuller = -8.2704, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

The null hypothesis non-starionarity can be rejected for upto 99% confidence as we can see from the p value from the ADF test; and for KPSS:

```
kpss.test(residuals(auto_select_AIC2))
```

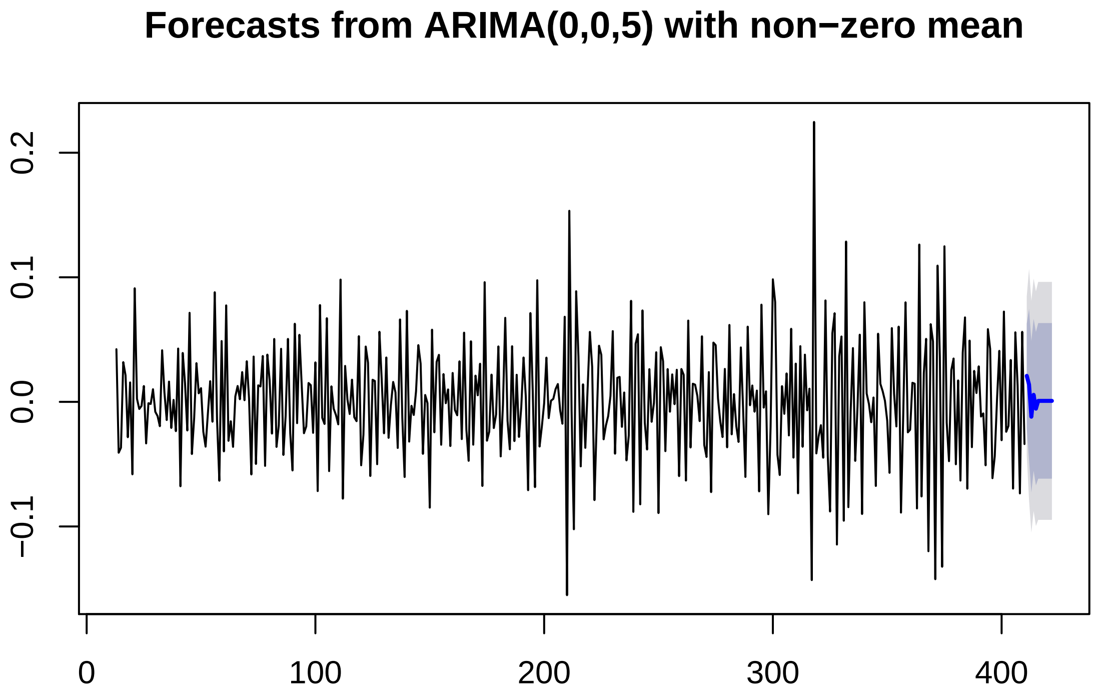
```
##
##  KPSS Test for Level Stationarity
##
## data:  residuals(auto_select_AIC2)
## KPSS Level = 0.59146, Truncation lag parameter = 4, p-value =
## 0.02341
```

Starionarity being null here, but it is still able to be rejected. The tests don't agree; and with the ACF graph as well, it is unlikely that the residual is stationary, thus it's not likely white noise.

Now we forecast:

d)

```
fcast2 <- forecast(auto_select_AIC2, h=12)
plot(fcast2)
```



e)

The point forecasts and the corresponding CI's (Lo 95, Hi 95) can be seen from the output :

```
summary(fcast2)
```

```
##
## Forecast method: ARIMA(0,0,5) with non-zero mean
##
## Model Information:
## Series: tsfinal
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##           ma1     ma2     ma3      ma4     ma5   mean
##       -1.0525  0.0381  0.1146  -0.2019  0.2492  8e-04
## s.e.   0.0488  0.0682  0.0698   0.0602  0.0499  2e-04
##
## sigma^2 estimated as 0.001067:  log likelihood=798.8
## AIC=-1583.61   AICc=-1583.32   BIC=-1555.7
##
## Error measures:
##                         ME       RMSE       MAE      MPE     MAPE
## Training set -0.0001254775 0.03241801 0.0252342 45.91441 150.5888
##                   MASE        ACF1
## Training set 0.3880047 0.008097829
##
```

```
## Forecasts:
##     Point Forecast       Lo 80      Hi 80       Lo 95      Hi 95
## 411   0.0208936356 -0.02096846 0.06275573 -0.04312892 0.08491619
## 412   0.0138248490 -0.04695157 0.07460127 -0.07912466 0.10677436
## 413  -0.0119161393 -0.07271348 0.04888120 -0.10489765 0.08106537
## 414   0.0056641553 -0.05532208 0.06665039 -0.08760624 0.09893455
## 415  -0.0054691096 -0.06703806 0.05609985 -0.09963070 0.08869248
## 416   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 417   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 418   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 419   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 420   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 421   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
## 422   0.0008080839 -0.06163812 0.06325428 -0.09469514 0.09631131
```

f)

For comparison, we need to look at the residuals of the original, unshortened data when we apply the same procedue of decomposition to it. We go through the same process as we did for the shortened data below:

```r
dat_df2 = ts_df(beer_dat_log) %>% rename(time = time,
                                             value = value)
dat_tbl2 = as_tbl_time(dat_df2,index=time)

trend_comp2 = ts_df(SMA(beer_dat_log,n=12)) %>% rename(time=time,SMA_12=value)


dat_tbl2  =  full_join(dat_tbl2,trend_comp2) %>%
                    mutate(SMA_resid2=beer_dat_log-SMA_12)
```

```r
SMA_resid_ts2 = ts(dat_tbl2%>% filter(!is.na(SMA_resid2)) %>% pull(SMA_resid2),
   start=c(1956,12),frequency=12)

season_comp2 = season(SMA_resid_ts2,d=12)

SMA_resid_tbl2 = ts_df(SMA_resid_ts2) %>% rename(time=time,
                                               SMA_resid2=value) %>%
                  mutate(seasonal2 = season_comp2) %>% as_tbl_time(index=time)

dat_tbl2 = full_join(dat_tbl2,SMA_resid_tbl2)
```

```r
dat_tbl2 = dat_tbl2 %>% mutate(deseason2=beer_dat_log-seasonal2)
```

```r
deseason_ts2 = ts(dat_tbl2 %>% filter(!is.na(deseason2)) %>% pull(deseason2),
                 start=c(1956,12),frequency=12)

deseason_trend2 = ts_df(SMA(deseason_ts2,n=2)) %>% rename(time=time,de_SMA_2 = value)

dat_tbl2 = full_join(dat_tbl2,deseason_trend2)
```

```r
dat_tbl2 = dat_tbl2 %>% mutate(Final_resid2=beer_dat_log-de_SMA_2-seasonal2)
```

```r
t_s2 = ts(dat_tbl2 %>% pull(Final_resid2))

tsfinal2 = na.remove(t_s2)
```

Thus the compared reidual values to the predicted vs original data is below:

```
vals2 <-(as.numeric(tail(tsfinal2,12)))
predicted2<-(as.numeric(fcast2$mean))
cbind(vals2,predicted2)
```

```
##               vals2    predicted2
##  [1,] -6.205617e-05  0.0208936356
##  [2,]  2.177917e-02  0.0138248490
##  [3,]  1.568507e-02 -0.0119161393
##  [4,] -1.249352e-03  0.0056641553
##  [5,]  3.746352e-03 -0.0054691096
##  [6,] -2.615302e-02  0.0008080839
##  [7,] -7.428512e-02  0.0008080839
##  [8,]  1.251300e-01  0.0008080839
##  [9,]  1.028108e-02  0.0008080839
## [10,] -6.989737e-02  0.0008080839
## [11,]  4.306215e-03  0.0008080839
## [12,] -1.765639e-02  0.0008080839
```

Now we display the forecast errors:

```
errors2<-abs(vals2-predicted2)
cbind(i,errors2)
```

```
##        i     errors2
##  [1,]  1 0.020955692
##  [2,]  2 0.007954321
##  [3,]  3 0.027601213
##  [4,]  4 0.006913507
##  [5,]  5 0.009215462
##  [6,]  6 0.026961106
##  [7,]  7 0.075093208
##  [8,]  8 0.124321933
##  [9,]  9 0.009472996
## [10,] 10 0.070705457
## [11,] 11 0.003498131
## [12,] 12 0.018464469
```

The value from the residue of the unshortened ts:

```
last_value2 <- tail(tsfinal2,1)
last_value2
```

```
## Time Series:
## Start = 422
## End = 422
## Frequency = 1
## [1] -0.01765639
## attr(,"na.removed")
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

-0.01765639 is indeed within the corresponding bounds of [-0.09469514,0.09631131].

Comparison of errors for the 2 methods:

```
cbind(errors,errors2)
```

```
##              errors     errors2
##  [1,] 0.0332810075 0.020955692
##  [2,] 0.0001652292 0.007954321
##  [3,] 0.0143288433 0.027601213
##  [4,] 0.0039560328 0.006913507
##  [5,] 0.0171674452 0.009215462
##  [6,] 0.0350132317 0.026961106
##  [7,] 0.1301927777 0.075093208
##  [8,] 0.0803641450 0.124321933
##  [9,] 0.0961208594 0.009472996
## [10,] 0.0222145871 0.070705457
## [11,] 0.0302733733 0.003498131
## [12,] 0.0496648714 0.018464469
```

Where "errors" corresponds to the forecast errors from model 1, and "errors2" from model 2. Let's see which one is higher on average:

```
cbind(mean(errors),mean(errors2))
```

```
##             [,1]       [,2]
## [1,] 0.04272853 0.03342979
```

The (absolute) errors for the second model appear to be lower, but let's take a look at the relative errors:

```
rerror1 <- abs((predicted-vals)/vals)
rerror2 <- abs((predicted2-vals2)/vals2)
cbind(rerror1,rerror2)
```

```
##            rerror1     rerror2
##  [1,] 6.533694e-03 337.6890900
##  [2,] 3.282035e-05   0.3652261
##  [3,] 2.849169e-03   1.7597120
##  [4,] 8.059993e-04   5.5336742
##  [5,] 3.434013e-03   2.4598492
##  [6,] 7.003715e-03   1.0308983
##  [7,] 2.660198e-02   1.0108781
##  [8,] 1.525855e-02   0.9935420
##  [9,] 1.800198e-02   0.9214009
## [10,] 4.204757e-03   1.0115610
## [11,] 5.936111e-03   0.8123447
## [12,] 9.938515e-03   1.0457672
```

As we can see, the relative (percentage) errors for the first model are much, much lower. Thus it's safe to say that the first approach made much less error, and it seems to be a vastly better fit than the second model.